

SVM实验报告

陈家豪 19307130210

一、数据处理

本次实验数据集为 Breast cancer dataset

通过sklearn.datasets获取数据

```
1 cancer = load_breast_cancer()
2 cancer_x=cancer.data
3 cancer_y=cancer.target
```

查看数据大小即部分数据：

```
1 print("加载完毕，数据大小：")
2 print(cancer_x.shape)
3 print(cancer_y.shape)
4 print("前5个数据：")
5 for i in range(5):
6     print(cancer_x[i],cancer_y[i])
```

```
开始加载数据
加载完毕，数据大小：
(569, 30)
(569,)
前5个数据：
[1.799e+01 1.038e+01 1.228e+02 1.001e+03 1.184e-01 2.776e-01 3.001e-01
 1.471e-01 2.419e-01 7.871e-02 1.095e+00 9.053e-01 8.589e+00 1.534e+02
 6.399e-03 4.904e-02 5.373e-02 1.587e-02 3.003e-02 6.193e-03 2.538e+01
 1.733e+01 1.846e+02 2.019e+03 1.622e-01 6.656e-01 7.119e-01 2.654e-01
 4.601e-01 1.189e-01] 0
[2.057e+01 1.777e+01 1.329e+02 1.326e+03 8.474e-02 7.864e-02 8.690e-02
 7.017e-02 1.812e-01 5.667e-02 5.435e-01 7.339e-01 3.398e+00 7.408e+01
 5.225e-03 1.308e-02 1.860e-02 1.340e-02 1.389e-02 3.532e-03 2.499e+01
 2.341e+01 1.588e+02 1.956e+03 1.238e-01 1.866e-01 2.416e-01 1.860e-01
 2.750e-01 8.902e-02] 0
[1.969e+01 2.125e+01 1.300e+02 1.203e+03 1.096e-01 1.599e-01 1.974e-01
 1.279e-01 2.069e-01 5.999e-02 7.456e-01 7.869e-01 4.585e+00 9.403e+01
 6.150e-03 4.006e-02 3.832e-02 2.058e-02 2.250e-02 4.571e-03 2.357e+01
 2.553e+01 1.525e+02 1.709e+03 1.444e-01 4.245e-01 4.504e-01 2.430e-01
```

breast cancer 的影响因素有30个，结果为0,1，表示是否患病

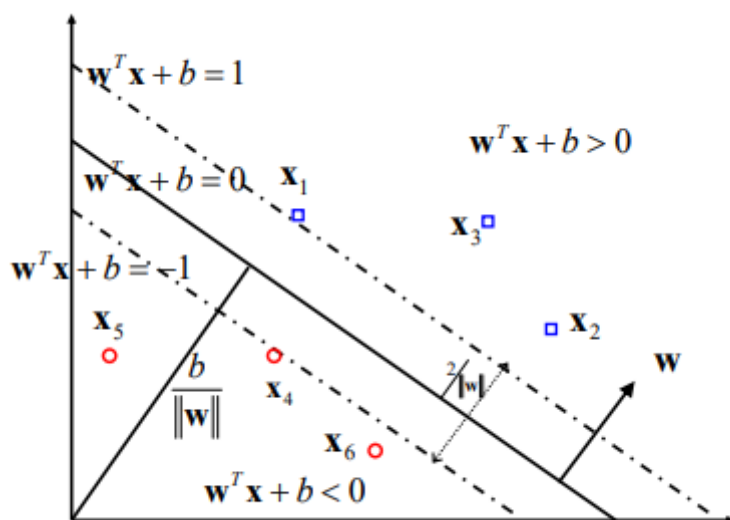
对数据进行拆分, test_size=0.2:

```
x_train,x_test,y_train, y_test=train_test_split(cancer_x,cancer_y,test_size=0.2)
```

二、算法

SVM算法原理

SVM学习的基本想法是求解能够正确划分训练数据集并且几何间隔最大的分离超平面。如下图所示即为分离超平面。



几何间隔：对于给定的数据集 和超平面，定义超平面关于样本点的几何间隔为

$$\gamma_i = y_i \left(\frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot \mathbf{x}_i + \frac{b}{\|\mathbf{w}\|} \right)$$

超平面关于所有样本点的几何间隔的最小值为 $\gamma = \min_{i=1,2,\dots,N} \gamma_i$

根据以上定义，SVM模型的求解最大分割超平面问题可以表示为以下约束最优化问题

$$\begin{aligned} \max_{\mathbf{w}, b} \quad & \gamma \\ \text{s.t.} \quad & y_i \left(\frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot \mathbf{x}_i + \frac{b}{\|\mathbf{w}\|} \right) \geq \gamma, i = 1, 2, \dots, N \end{aligned}$$

经变化，SVM模型的求解最大分割超平面问题又可以表示为以下约束最优化问题

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, i = 1, 2, \dots, N \end{aligned}$$

对其使用拉格朗日乘子法得到其对偶问题（dual problem）。将有约束的原始目标函数转换为无约束的新构造的拉格朗日目标函数

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^N \alpha_i (y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1)$$

经变化得：

$$\begin{aligned} L(\mathbf{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) - \sum_{i=1}^N \alpha_i y_i \left(\left(\sum_{j=1}^N \alpha_j y_j \mathbf{x}_j \right) \cdot \mathbf{x}_i + b \right) + \sum_{i=1}^N \alpha_i \\ &= -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) + \sum_{i=1}^N \alpha_i \end{aligned}$$

即

$$\min_{\mathbf{w}, b} L(\mathbf{w}, b, \boldsymbol{\alpha}) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) + \sum_{i=1}^N \alpha_i$$

$$\text{得到最优解 } \boldsymbol{\alpha}^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_N^*)^T$$

计算：

$$\mathbf{w}^* = \sum_{i=1}^N \alpha_i^* y_i \mathbf{x}_i$$

选择 $\boldsymbol{\alpha}^*$ 的一个分量 α_j^* 满足条件 $0 < \alpha_j^* < C$ ，计算

$$b^* = y_j - \sum_{i=1}^N \alpha_i^* y_i (\mathbf{x}_i \cdot \mathbf{x}_j)$$

求分离超平面： $\mathbf{w}^* \cdot \mathbf{x} + b^* = 0$

kernel

用核函数替换目标函数和分类决策函数中的内积。核函数表示，通过一个非线性转换后的两个实例间的内积。

$$K(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{z})$$

求分离超平面

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^N \alpha_i$$

$$s.t. \quad \sum_{i=1}^N \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, N$$

$$\text{得到最优解 } \alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_N^*)^T$$

分类决策函数

$$f(x) = \text{sign} \left(\sum_{i=1}^N \alpha_i^* y_i K(x, x_i) + b^* \right)$$

核函数有三种：

1.高斯核函数 (rbf)

$$k(x, x') = e^{-\frac{\|x - x'\|^2}{2\sigma^2}}$$

2.多项式核函数

$$K(x, y) = (x \cdot y + c)^d$$

3.sigmoid核函数

$$K(x, z) = \tanh(\gamma x \bullet z + r)$$

线性核函数就是 $K(x, z) = x \bullet z$

三、模型

获取数据后选取各类模型，用训练集数据训练，然后用模型预测测试集结果，与真实结果比较，计算各模型的正确率。

第一个模型：linear kernel

```
1 svc_clf = SVC(kernel='linear')
2 svc_clf.fit(x_train,y_train)
3 y_pred = svc_clf.predict(x_test)
4 print("正确率为: %f"% accuracy(y_test,y_pred))
```

第二个模型：poly kernel

```
1 svc_clf2 = SVC(kernel='poly', degree=4)
2 svc_clf2.fit(x_train,y_train)
3 y_pred2 = svc_clf2.predict(x_test)
4 print("正确率为: %f"%accuracy(y_test,y_pred2))
```

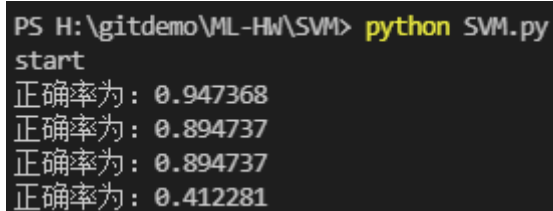
第三个模型：rbf kernel

```
1 svc_clf3 = SVC(kernel='rbf')
2 svc_clf3.fit(x_train,y_train)
3 y_pred3 = svc_clf3.predict(x_test)
4 print("正确率为: %f"%accuracy(y_test,y_pred3))
```

第四个模型：sigmoid kernel

```
1 svc_clf4 = SVC(kernel='sigmoid')
2 svc_clf4.fit(x_train,y_train)
3 y_pred4 = svc_clf4.predict(x_test)
4 print("正确率为: %f"%accuracy(y_test,y_pred4))
```

四、结果



```
PS H:\gitdemo\ML-HW\SVM> python SVM.py
start
正确率为: 0.947368
正确率为: 0.894737
正确率为: 0.894737
正确率为: 0.412281
```

在4个模型中

线性的正确率最大,为0.947

sigmoid的正确率最小, 为0.412

poly,rbf kernel 的正确率相同, 为0.895

本次实验中linear kernel效果最好